Methods for rapidly processing angular masks of next-generation galaxy surveys

Molly E. C. Swanson,[1]⋆ Max Tegmark,[1] Andrew J. S. Hamilton,[2] and J. Colin Hill[1]

[1]*Dept. of Physics and MIT Kavli Institute, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139,USA*
[2]*JILA and Dept. of Astrophysical and Planetary Sciences, Box 440, University of Colorado, Boulder, CO 80309, USA*

ABSTRACT

As galaxy surveys become larger and more complex, keeping track of the completeness, magnitude limit, and other survey parameters as a function of direction on the sky becomes an increasingly challenging computational task. For example, typical angular masks of the Sloan Digital Sky Survey contain about $N = 300,000$ distinct spherical polygons. Managing masks with such large numbers of polygons becomes intractably slow, particularly for tasks that run in time $\mathcal{O}\left(N^2\right)$ with a naive algorithm, such as finding which polygons overlap each other. Here we present a "divide-and-conquer" solution to this challenge: we first split the angular mask into predefined regions called "pixels," such that each polygon is in only one pixel, and then perform further computations, such as checking for overlap, on the polygons within each pixel separately. This reduces $\mathcal{O}\left(N^2\right)$ tasks to $\mathcal{O}\left(N\right)$, and also reduces the important task of determining in which polygon(s) a point on the sky lies from $\mathcal{O}\left(N\right)$ to $\mathcal{O}\left(1\right)$, resulting in significant computational speedup. Additionally, we present a method to efficiently convert any angular mask to and from the popular *HEALPix* format. This method can be generically applied to convert to and from any desired spherical pixelization. We have implemented these techniques in a new version of the MANGLE software package, which is freely available at `http://space.mit.edu/home/tegmark/mangle/`, along with complete documentation and example applications. These new methods should prove quite useful to the astronomical community, and since MANGLE is a generic tool for managing angular masks on a sphere, it has the potential to benefit terrestrial mapmaking applications as well.

Key words: large-scale structure of Universe – methods: data analysis – surveys

1 INTRODUCTION

Over the past few decades, galaxy surveys have provided a wealth of information about the large-scale structure of our Universe, and the next generation of surveys currently being planned promises to provide even more insight. In order to realize the full potential of upcoming surveys, it is essential to avoid unnecessary errors and approximations in the way they are analysed. The tremendous volumes of data produced by these new surveys will shrink statistical uncertainty to unprecedented levels, and in order to take advantage of this we must ensure that the systematic uncertainties can keep pace. The purpose of this paper is to maximize the scientific utility of next-generation surveys by providing methods for processing angular masks as rapidly and accurately as possible.

Angular masks of a galaxy survey are functions of direction on the sky that model the survey completeness, magnitude limit, seeing, dust extinction, or other parameters that vary across the sky. The earliest galaxy redshift surveys – the first Center for Astrophysics redshift survey (CfA1; Huchra et al. 1983) and the

first Southern Sky Redshift Survey (SSRS1; da Costa et al. 1991) – had simple angular masks defined by boundaries in declination and Galactic latitude. The next generation of surveys – IRAS (Strauss et al. 1992) and PCSz (Saunders et al. 2000) – had somewhat more complex masks, with some regions of high contamination excluded from the survey.

The present generation of surveys – the Two Degree Field Galaxy Redshift Survey (2dFGRS; Colless et al. 2001, 2003) and the Sloan Digital Sky Survey (SDSS; York et al. 2000) – consist of photometric surveys that identify galaxies and measure their angular positions combined with spectroscopic surveys that measure a redshift for each galaxy to determine its distance from us. Angular masks are useful for describing parameters for both photometric and spectroscopic surveys – for example, seeing and magnitude limit are key parameters to model in photometric surveys, and the survey completeness – i.e., the fraction of photometrically selected target galaxies for which a spectrum has been measured – is vital for analysing spectroscopic surveys.

The angular masks of SDSS and 2dFGRS consist of circular fields defined by the spectroscopic plates of the redshift survey superimposed on an angular mask of the parent photometric sur-

⋆ E-mail: molly@space.mit.edu
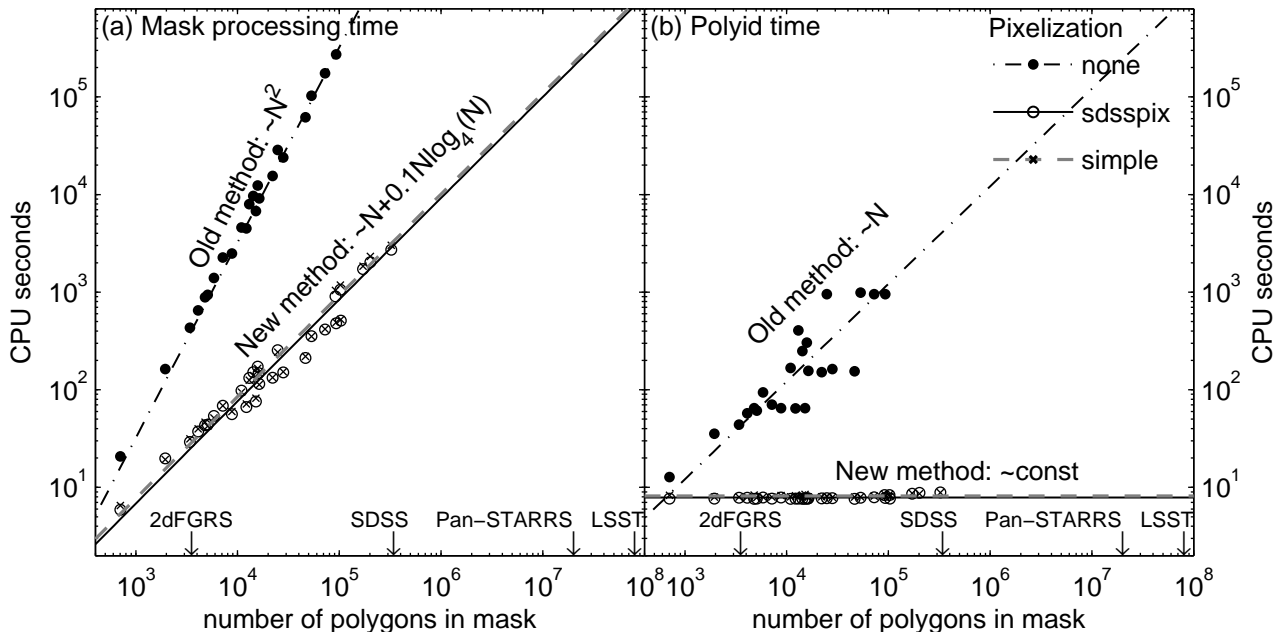
Figure 1. Speed trials for a series of portions of the SDSS DR5 mask with and without pixelization. (a) Time required for pixelization, snapping, balkanization and unification of the mask. (b) Time required to identify in which polygon each of the ∼400,000 SDSS DR5 galaxies lies. Each set of trials is fitted with a power law to show how the processing time scales with the number of polygons N. Also shown on the x-axis are the number of polygons in the 2dFGRS mask, the SDSS DR5 mask, and conservative estimates for the Pan-STARRS and LSST large-scale structure masks based on scaling up 2dFGRS.

vey. 2dFGRS uses the Automatic Plate Measurement (APM) survey (Maddox et al. 1990a,b, 1996) as its parent photometric survey and covers approximately 1500 square degrees. The APM angular mask consists of 269 $5° \times 5°$ photographic plates which are drilled with numerous holes to avoid bright stars, satellite trails, plate defects, and so forth. Combining the photographic plates, holes, and spectroscopic fields gives a total of 3525 polygons that define the spectroscopic angular mask of 2dFGRS.

The SDSS covers a larger area on the sky – 5740 square degrees in Data Release 5 (DR%; Adelman-McCarthy et al. 2007) – and has a yet more complicated angular mask than 2dFGRS. The SDSS photometric survey is done by drift-scanning: each scan across the sky covers six long, narrow scanlines, and the gaps between these lines are filled in with a second scan slightly offset from the first, producing a "stripe" about $2.5°$ wide assembled from 12 scanlines. In addition to the fairly intricate pattern produced by this scanning strategy, there are nearly 250,000 holes masked out of the photometric survey for various reasons, plus the circular $3°$ spectroscopic fields. Combining all of these elements produces an angular mask for the spectroscopic survey that contains 340,351 polygons.

To accurately manage the 2dFGRS and SDSS angular masks, Hamilton & Tegmark (2004) developed a suite of general-purpose software called MANGLE, which performs several important procedures on angular masks using computational methods detailed in Hamilton (1993a,b). This software has proved to be a valuable resource to the astronomical community: it has been used in several analyses of galaxy survey data (Tegmark et al. 2002, 2004; Mandelbaum et al. 2005; Hikage et al. 2005; Park et al. 2005; Conroy et al. 2005; Park et al. 2007; Nishimichi et al. 2007; Shen et al. 2007; Wang et al. 2007; Tinker et al. 2007). Additionally, it was used extensively in the preparation of the New York University Value-Added Galaxy Catalog (NYU-VAGC; Blanton et al.

2005), which has been used as the basis for almost all publications on large-scale structure by the SDSS collaboration.

However, many of functions in the original version of MAN-GLE run in $\mathcal{O}\left(N^2\right)$ time, which becomes quite computationally challenging as the size and complexity of surveys continues to increase – computations involving the SDSS mask can take several months of CPU time. In this paper we present new algorithms that can process complicated angular masks such as SDSS dramatically faster with no loss of accuracy. Our method is based on splitting an angular mask into pixels, reducing the processing time to $\mathcal{O}\left(N\right)$ by adding an $\mathcal{O}\left(N \log N\right)$ preprocessing step. Similar methods based on hierarchical spatial subdivisions have been found to be useful in the field of computational geometry (see, e.g., Goodman & O'Rourke 2004), but have not previously been applied to angular masks in an astronomy context.

This will be especially useful for next-generation surveys, such as the Dark Energy Survey (DES; Dark Energy Survey Collaboration 2005) and surveys done with the Wide-Field Multi-Object Spectrograph (WFMOS; Yamamoto et al. 2006; Glazebrook et al. 2006), the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS; Kaiser 2004), and the Large Synoptic Survey Telescope (LSST; Tyson 2002; Stubbs et al. 2004; Tyson 2006; LSST Collaboration 2006). DES, Pan-STARRS, and LSST will perform photometric surveys and use techniques for estimating redshifts based on the photometric information; WFMOS will perform spectroscopic surveys using one of the upcoming photometric surveys for target selection. The methods we present here are useful for both photometric and spectroscopic surveys – keeping track of factors such as seeing and dust extinction could prove particularly important for photometric redshift determinations (Collister et al. 2007; Oyaizu et al. 2008; Banerji et al. 2008).

The proposed large-scale structure survey to be produced by
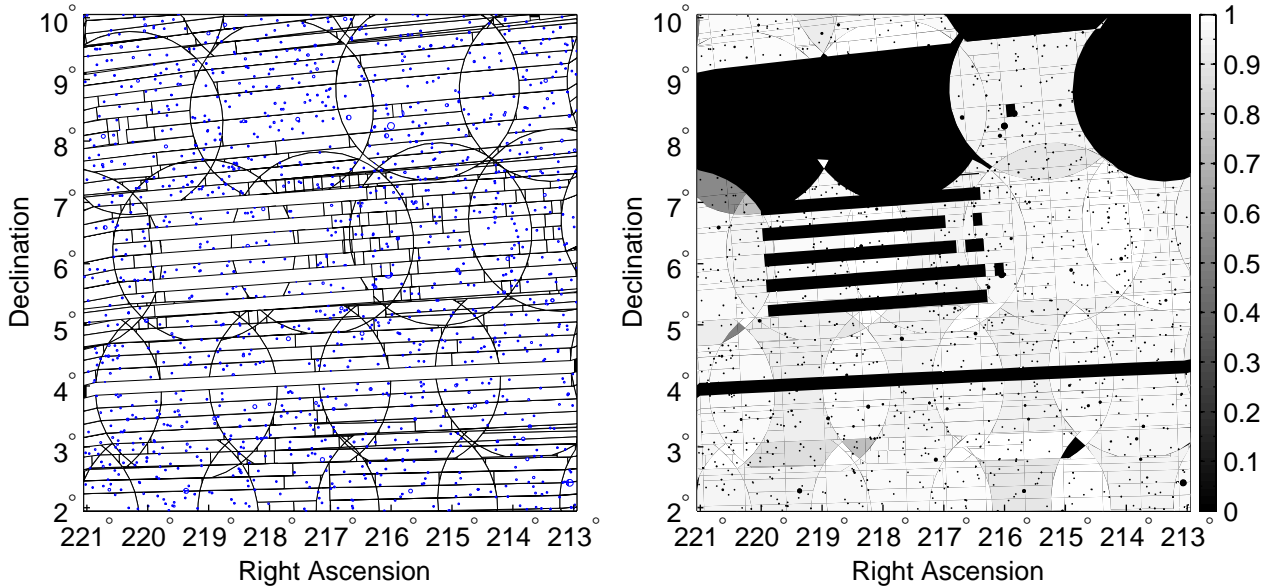
Figure 2. A portion of the SDSS DR5 angular mask (Blanton et al. 2005; Adelman-McCarthy et al. 2007). Left: Polygons defining the mask: spectroscopic plates and lines delineating different scans and spectroscopic plates are shown in black, and holes in the mask are shown in blue/gray. Right: Processed version of the mask, shaded according to survey completeness.

**Table 1.** Definitions of Terms, in Alphabetical Order

Term	Definition
boundary	A set of edges bounding a polygon.
cap	A spherical disk, a region above a circle on the unit sphere.
circle	A line of constant latitude with respect to some arbitrary polar axis on the unit sphere.
edge	An edge is part of a circle. A polygon is enclosed by its edges.
great circle	A line of zero latitude with respect to some arbitrary polar axis on the unit sphere. A great circle is a circle, but a circle is not necessarily a great circle.
mask	The union of an arbitrary number of weighted polygons.
pixel	A special polygon that specifies some predefined region on the sky as part of a scheme for discretizing the unit sphere. Once a mask has been pixelized, each polygon in the mask is guaranteed to overlap with exactly one pixel.
polygon	The intersection of an arbitrary number of caps.
rectangle	A special kind of polygon, a rectangular polygon bounded by lines of constant longitude and latitude.
vertex	A point of intersection of two circles. A vertex of a polygon is a point where two of its edges meet.
weight	The weight assigned to a polygon. The spherical harmonics of a mask are the sum of the spherical harmonics of its polygons, each weighted according to its weight. A weight of 1 is the usual weight. A weight of 0 signifies an empty polygon, a hole. In general the weight may be some arbitrary positive or negative real number.

Pan-STARRS will cover ∼30,000 square degrees in 5 wavelength bands, with each field being observed ∼50 times such that the images can be co-added. A naive scaling up of the 2dFGRS area and number of polygons gives an estimate of $\sim 2 \times 10^7$ polygons for the final Pan-STARRS mask. Similarly, the LSST large-scale structure survey will cover ∼20,000 square degrees in 6 bands, with ∼200 co-added images, suggesting $\sim 8 \times 10^7$ polygons for the LSST mask. The need for an improvement in mask processing speed is clearly illustrated in Fig. 1: with the old algorithms, the projected processing time for the LSST mask would be over 6000 years. With our new method, this time is reduced by a factor of ∼24,000 to just ten days.

In addition to developing faster algorithms for processing angular masks, we have also integrated the MANGLE utilities with *HEALPix*, a widely used tool for discretizing the celestial sphere

(Górski et al. 2005). The methods used by MANGLE are complementary to *HEALPix*: MANGLE is best used for functions that are piecewise-constant in distinct regions of the sky, such as the completeness of a galaxy survey. In contrast, *HEALPix* is optimal for describing functions that are continuously varying across the sky, such as the cosmic microwave background (CMB) or the amount of extinction due to Galactic dust. The ability to convert rapidly between these two formats allows for easy comparison of these two types of data without the unnecessary approximation inherent in discretizing an angular mask. Furthermore, converting a mask into *HEALPix* format allows users to take advantage of pre-existing *HEALPix* tools for rapidly computing spherical harmonics.

The spectacular surveys on the horizon are preparing to generate massive, powerful datasets that will be made publically available – this in turn necessitates powerful and intuitive general-
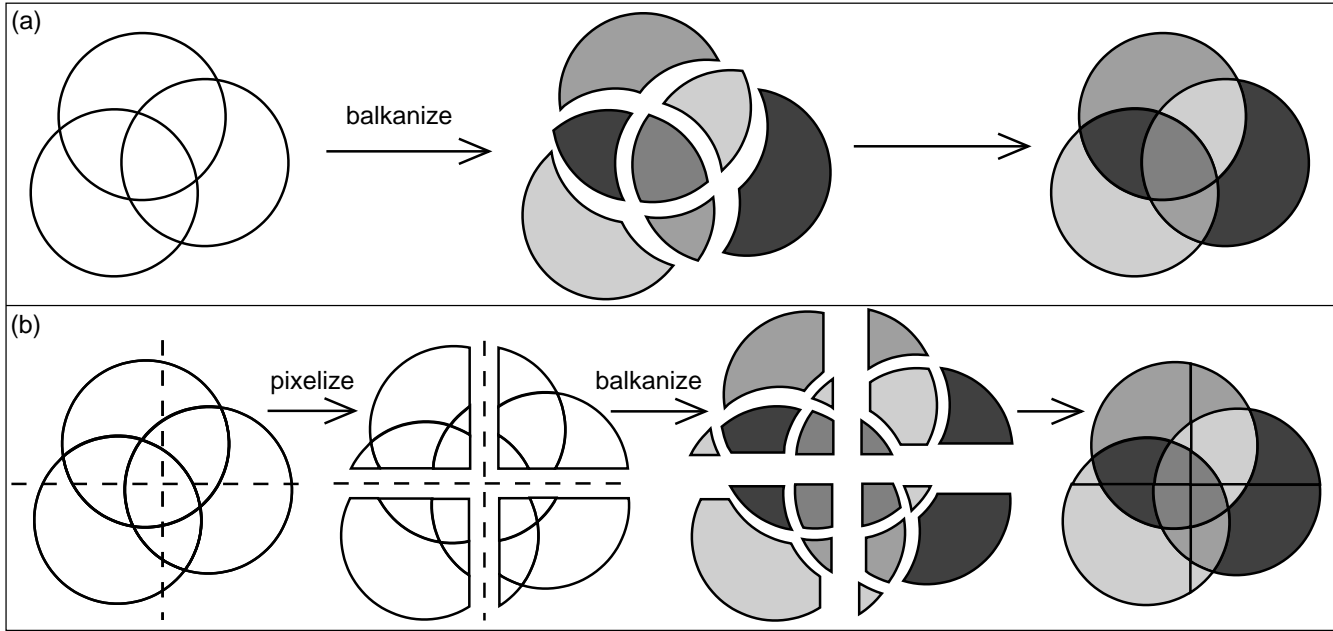
Figure 3. A cartoon illustrating the process of balkanization (a) with no pixelization, (b) with pixelization.

purpose tools that assist the community to do science with this avalanche of data. We provide a such tools with this new generation of the MANGLE software and describe these new tools here. However, this paper is not a software manual (a manual is provided on the MANGLE website) but rather a description of the underlying algorithms. These tools have been utilized in recent analyses of SDSS data (Tegmark et al. 2006; Swanson et al. 2008); we are now making them public so others can use them as well.

The outline of this paper is as follows: in §2 we give an overview of the terminology we use to describe angular masks and the basic tasks we wish to perform, and in §3 we detail our algorithms for accelerating these tasks and quantify their speed. We describe our methods for integrating MANGLE with *HEALPix* in §4 and summarize in §5.

2 MANGLE TERMINOLOGY

The process of defining an angular mask of a galaxy survey in a generic way requires a set of standardized terminology. We use the terminology from Hamilton & Tegmark (2004) and present a summary of it here. Our formal definition of an angular mask is a union of an arbitrary number of weighted angular regions bounded by arbitrary numbers of edges. The restrictions on the mask are

(i) that each edge must be part of some circle on the sphere (but not necessarily a great circle), and

(ii) that the weight within each subregion of the mask must be constant.

This definition does not cover every theoretical possibility of how a piecewise-constant function on a sphere could be defined, but in practice it is sufficiently broad to accommodate the design of essentially any galaxy survey. Furthermore, as we discuss in detail in §4, a curvilinear angular region (such as a *HEALPix* pixel) can be well-approximated by segments of circles at high resolution. As an example of a typical angular mask, we show a portion of the SDSS

angular mask from Data Release 5 (DR5; Adelman-McCarthy et al. 2007) in Fig. 2.

The fundamental building block of an angular mask is the spherical polygon, which is defined as a region bounded by edges that are part of a circle on the sphere. An angular mask is thus the union of arbitrarily weighted non-overlapping polygons. For convenience, we provide an updated version of a table from Hamilton & Tegmark (2004) in Table 1 containing the definitions of key terms used in this paper.

The basic procedure used by the MANGLE software to process an angular mask consists of the following steps, which are described in greater detail in Hamilton & Tegmark (2004):

(i) Snap
(ii) Balkanize
(iii) Weight
(iv) Unify

The snapping step identifies edges of polygons that are nearly coincident and snaps them together so the edges line up exactly. This is necessary because nearly-coincident edges can cause significant numerical issues in later computations. In practice, this situation occurs when two polygons in a survey are intended to abut perfectly, but are prevented from doing so by roundoff errors or numerical imprecision in the mask definition. There are several tunable tolerances that control how close two edges can be before they get snapped together – these can be adjusted based on how precisely the polygons defining the mask are specified.

Balkanization is the process of resolving a mask into a set of non-overlapping polygons. It checks for overlap between each pair of polygons in the mask, and if the polygons overlap, it fragments them into non-overlapping pieces. After this is completed, it identifies polygons having disconnected geometry and subdivides them into connected parts. The basic concept of balkanization is illustrated in the top panel of Fig. 3. The purpose of this procedure is to define all of the distinct regions on the sky in which the piecewise-constant function we are intending to model might take on a different value. For example, the 2dFGRS and SDSS spectroscopic

surveys generate masks containing many overlapping circles defining each spectroscopic field observed. The SDSS spectrographs can observe 640 objects in each field, so if there are more than 640 desired target galaxies in the field, they might not all be observed. For example, one field may have spectra for 80 per cent of the targets, and a neighbouring field may have 90 per cent, but in the region where they overlap all of the targets may have been observed. This is how the survey completeness is determined and illustrates why balkanization is necessary.

After the mask is balkanized, weights are assigned to each polygon, representing the value of the survey completeness (or any other desired parameter) in that region. The way this is done depends on how this information is provided for a given survey. For example, the 2dFGRS mask software by Peder Norberg and Shaun Cole[1] provides a function that takes an angular position on the sky and returns the completeness, the magnitude limit, the photographic plate number, and the value of the parameter μ (described in Colless et al. 2001) at that location. This information can be imported into MANGLE by producing a list of the midpoints of each of the polygons in the mask, applying the 2dFGRS software to calculate the value of the desired function at each midpoint, and assigning these weights to the appropriate polygon by using the "weight" routine in MANGLE. For the SDSS mask, the files provided by the NYU-VAGC[2] (Blanton et al. 2005) already include weights for the survey completeness, so this step is not needed.

The final step of the processing is unification, which discards polygons with zero weight and combines neighbouring polygons that have the same weight. While not strictly necessary, this procedure clears out unneeded clutter and makes subsequent calculations more efficient.

After an angular mask has been processed in this fashion, it can be used for function evaluation: i.e., given a point on the sky, determine in which one of the non-overlapping polygons it lies, and then get the weight of that polygon to obtain the value of the function at the input point. It can also be used for creating a random sample of points with the same selection function as the survey, calculating Data-Random $\langle DR \rangle$ and Random-Random $\langle RR \rangle$ angular integrals, and computing the spherical harmonics of the mask. The MANGLE software provides utilities for all of these tasks.

3 SPEEDUP: PIXELIZATION

The tasks of snapping, balkanization, and unification all require comparing pairs of polygons in the mask. The brute-force method to accomplish this is simply to compare each polygon with every other polygon, which is what the original version of MANGLE did. This naive algorithm is $\mathcal{O}\left(N^2\right)$, which is easily sufficient for masks such as the 2dFGRS mask, with an N of a few thousand polygons. However, the SDSS mask has about 100 times as many polygons, and points to the need for a cleverer approach. The method we present here is a divide-and-conquer approach we dub "pixelization," which processes the mask so that each polygon needs to be compared with only a few nearby polygons.

3.1 Pixelization concept

The underlying concept of pixelization is as follows: before performing any snapping, balkanization, or unification, divide the

mask into predefined regions called "pixels" and split each polygon along the pixel boundaries such that each polygon is only in one pixel. Then for following tasks, polygons need only be compared with other polygons in the same pixel.

The process of balkanization with and without pixelization is illustrated in Fig. 3 for three overlapping polygons. The top panel shows the unpixelized version: balkanization checks for overlap between each pair of polygons, and then fragments it into seven non-overlapping polygons.

The bottom panel shows the same process but using pixelization as a first step. First, each polygon is divided along the pixel boundaries, shown by the dotted lines. At this point, each of the four pixels shown has three polygons in it: the intersections between that pixel and each of the original three polygons. Then balkanization is performed within each pixel: the three polygons in the upper left pixel are split into five non-overlapping polygons, and so forth, yielding a final set of 18 non-overlapping polygons. In this illustrative example, pixelization increases the complexity of the process, but in general it replaces the $\mathcal{O}\left(N^2\right)$ algorithm for balkanization with one that is roughly $\mathcal{O}\left(M\left(N/M\right)^2\right)$ for N polygons and M pixels, which is roughly $\mathcal{O}\left(N\right)$ if $M \sim N$. For large, complicated masks such as SDSS, this speeds up the processing time by a factor of ~ 1200.

Once a mask has been pixelized, the important task of determining in which polygon(s) a given point lies is sped up greatly as well: one merely has to calculate in which pixel the point lies, and then test if the point is in each polygon within that pixel. For a typical pixelization scheme, the appropriate pixel number for a given point can be found with a simple formula, i.e. an $\mathcal{O}\left(1\right)$ calculation, so pixelization reduces the $\mathcal{O}\left(N\right)$ algorithm of testing every polygon in the mask to $\mathcal{O}\left(N/M\right)$. This means that with pixelization, this task does not depend on the total number of polygons in the mask at all if $M \sim N$.

It is important to note that the pixelization procedure makes no approximations to the original mask – the pixels are used simply as a tool to determine which polygons are close to each other, not as a means of discretizing the mask itself. A mask that has been balkanized after pixelization contains all of the same information as it would without pixelization, except that it took a tiny fraction of the time to produce. Furthermore, unification can be applied across the whole mask rather than within each pixel, which effectively unpixelizes the mask if desired. Thus there are essentially no drawbacks to using our pixelization procedure.
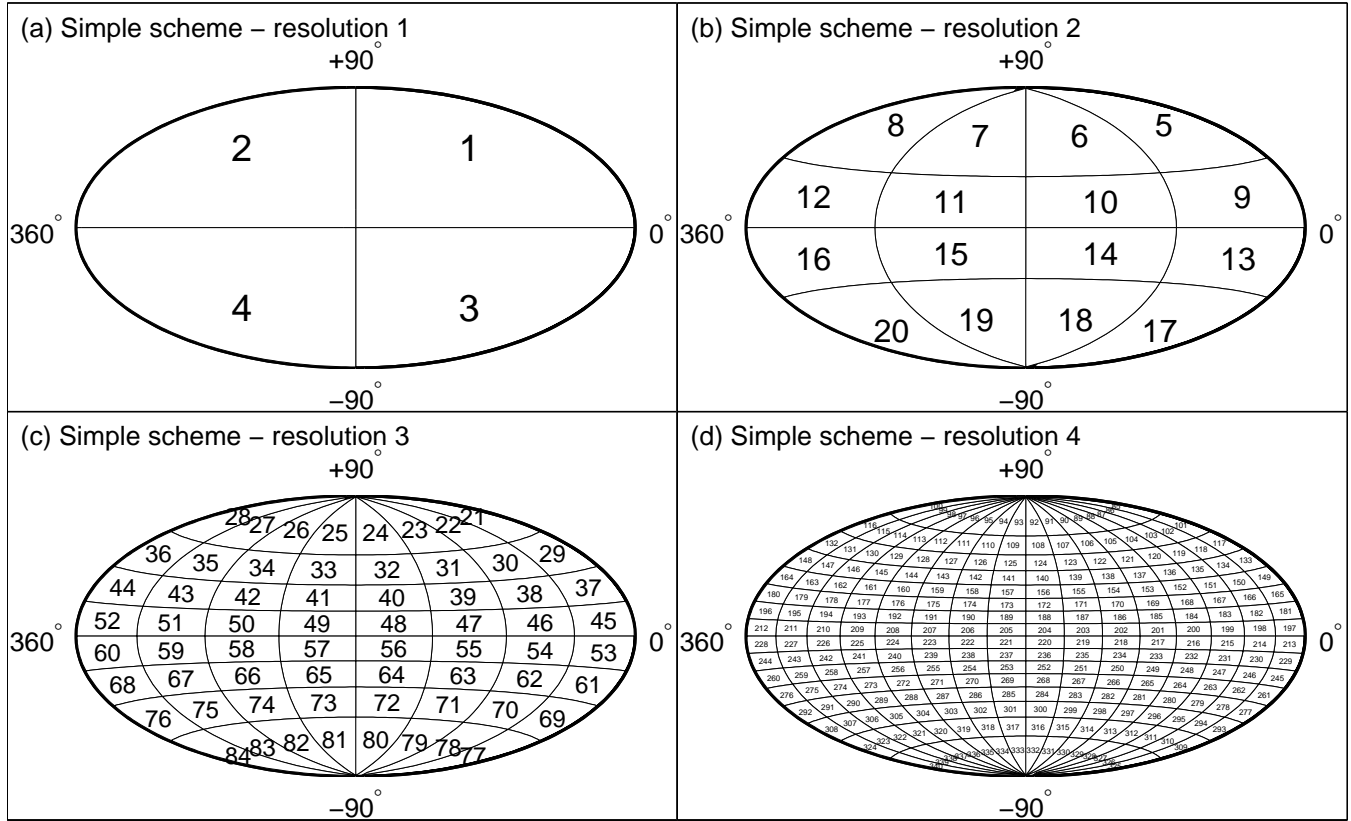
3.2 Pixelization schemes

3.2.1 Simple scheme

The most straightforward means of pixelizing the sky is to use lines of equal azimuth and elevation as the pixel boundaries. The azimuth and elevation typically correspond to celestial coordinates – right ascension (RA) and declination (Dec) – in a survey mask. In this scheme, the whole sky is split into quadrants along the equator and prime meridian to form the lowest resolution of pixelization. In MANGLE this is defined as resolution 1.

To pixelize to higher resolutions, each pixel is split into four child pixels, with the boundaries at the midpoints of azimuth and of cos(elevation) within the pixel. This creates pixels with equal area. Thus resolution 2 consists of each of the four resolution 1 quadrants split into four pixels each, and so forth. This procedure produces a hierarchical pixel structure known in computer science terminology as a quadtree (de Berg et al. 2000): there are 4^r pixels

Figure 4. The full sky (shown in a Hammer-Aitoff projection in celestial coordinates) pixelized with the simple pixelization scheme at the four lowest resolutions.

in this scheme at resolution r, and each of these pixels has four child pixels at resolution $r + 1$ and r parent pixels, one at each lower resolution. Resolution 0 is defined as being the whole sky.

In MANGLE, the pixels of the simple scheme are numbered as follows: the whole sky is pixel 0, the four quadrants of resolution 1 are pixels 1, 2, 3, and 4, resolution 2 is pixels 5-20, and so forth. Thus just one number specifies both the resolution and the pixel location. At each resolution, the pixels are numbered in a ring pattern, starting from an elevation of $90°$ and along increasing azimuth for each ring of equal elevation. The simple scheme pixels at the four lowest resolutions are shown in Fig. 4.

3.2.2 SDSSPix scheme

Alternatively, the pixelization can be done such that it is more closely aligned with the mask of a given survey. In particular, a pixelization scheme called SDSSPix[3] has been developed for use with the SDSS geometry. Like the simple scheme, SDSSPix is a hierarchical, equal-area pixelization scheme, and it is based on the SDSS survey coordinates λ and η. As described in Stoughton et al. (2002), SDSS survey coordinates form a spherical coordinate system rotated relative to the celestial coordinate system. The poles are located at $\mathrm{RA} = 95°, \mathrm{Dec} = 0°$ and $\mathrm{RA} = 275°, \mathrm{Dec} = 0°$ (J2000), which are strategically located outside the SDSS covered area and in the Galactic plane. η is the azimuthal angle, with lines

[3] http://lahmu.phyast.pitt.edu/~scranton/SDSSPix/

of constant η being great circles perpendicular to the survey equator, and λ is the elevation angle, with lines of constant λ being small circles parallel to survey equator. $\lambda = 0°, \eta = 0°$ is located at $\mathrm{RA} = 185°, \mathrm{Dec} = 32.5°$ with η increasing northward. This configuration has been chosen such that the stripes produced by the SDSS scanning pattern lie along lines of constant η.

The SDSSPix base resolution is defined by 36 divisions in the η direction (equally spaced in η) and 13 in the λ direction (equally spaced in $\cos \lambda$), for a total of 468 equal-area pixels. These divisions are chosen such that at a special resolution level (called the "superpixel" resolution), there is exactly one pixel across each SDSS stripe. Finally, as in the simple scheme, higher resolutions are achieved by hierarchically subdividing each pixel at a given resolution into four smaller pixels.

SDSSPix has been included in MANGLE by incorporating several routines from the SDSSPix software package available online.[3] The numbering scheme in the MANGLE implementation of SDSSPix differs somewhat from the internal SDSSPix numbering: in MANGLE, the entire sky is pixel 0, as in the simple scheme, but it has 117 child pixels (instead of 4), numbered from 1 to 117. These pixels, which comprise resolution 1, are not part of the official SDSSPix scheme, but are created by combining sets of 4 pixels from what is defined as resolution 2 in MANGLE. The resolution 2 pixels are the official SDSSPix base resolution pixels, and are numbered from 118 to 585. Higher resolutions are constructed through the standard SDSSPix hierarchical division of each pixel into 4 child pixels. As in the simple scheme, the pixel number identifies both the resolution and the pixel position. The superpixel res-
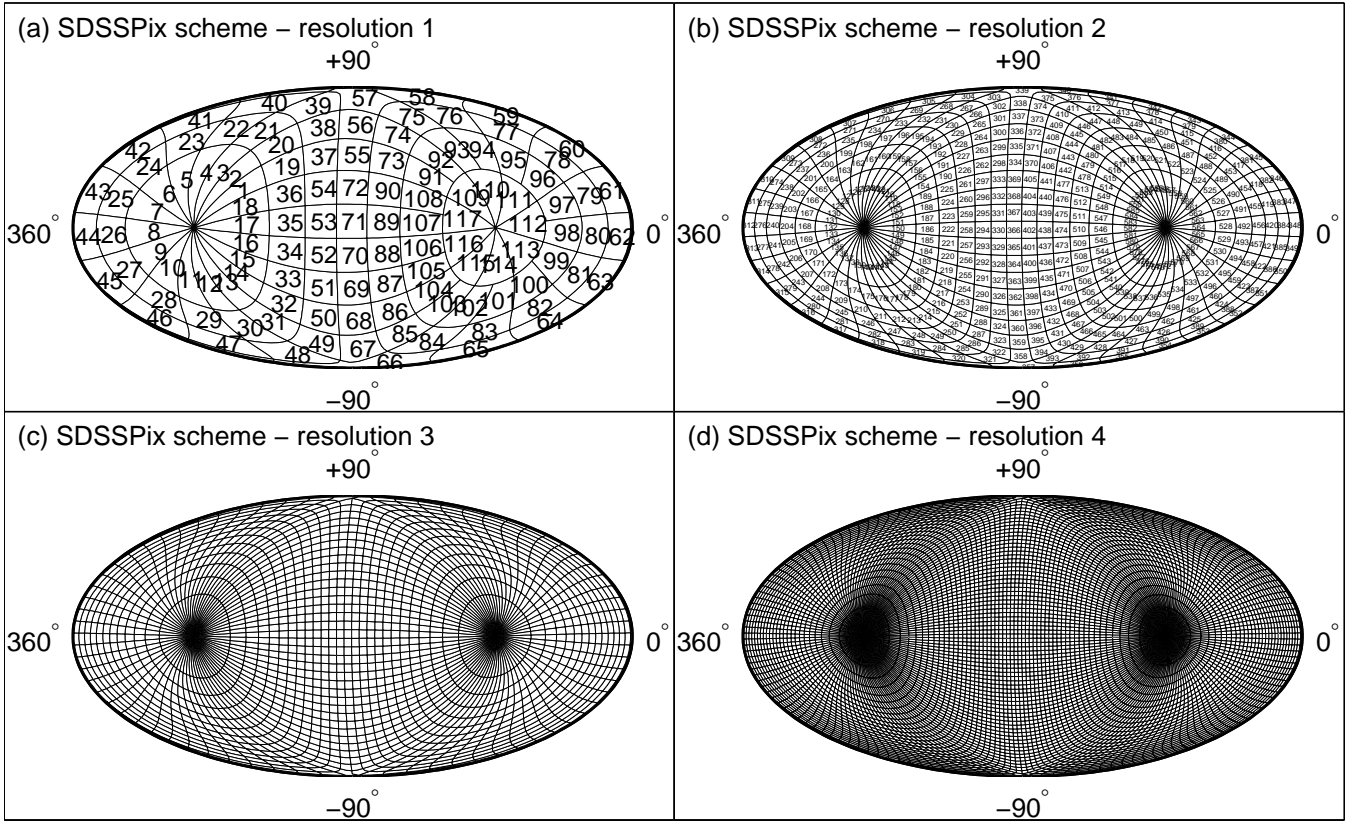
Figure 5. The full sky (shown in a Hammer-Aitoff projection in celestial coordinates) pixelized with the SDSSPix pixelization scheme at the four lowest resolutions.

olution described above is defined as resolution 5 in MANGLE and contains a total of 7488 pixels.

MANGLE typically uses RA and Dec as its internal azimuth and elevation coordinates, so the SDSSPix pixels are constructed as rectangles in $\eta - \lambda$ coordinates and then rotated into celestial coordinates. The SDSSPix pixels at the four lowest resolutions are shown in Fig. 5.

3.2.3 Other schemes

The implementation of pixelization in MANGLE is designed to be flexible: it is simple for users to add their own scheme as well. The pixelization uses only four basic routines:

(i) get_pixel: Given a pixel number, return a polygon representing that pixel.

(ii) which_pixel: Given a point on the sky and a resolution, return the pixel number containing the point.

(iii) get_child_pixels: Given a pixel number, return the numbers of its child pixels.

(iv) get_parent_pixels: Given a pixel number, return the numbers of its parent pixels.

Adding a new pixelization scheme simply requires creating appropriate versions of these four routines. Note that it also requires that the pixels can be represented as polygons – this is *not* strictly the case for the *HEALPix* pixels, as discussed further in §4.

3.3 Pixelization algorithm

The purpose of pixelization is to speed up the processing of angular masks, which means that the pixelization itself must be done with a clever, speedy algorithm or nothing will be gained. The naive algorithm is to search through all of the polygons in the mask for those that overlap that pixel. This is $\mathcal{O}(NM)$ for N polygons and M pixels, and is not sufficient for our purposes.

Our fast pixelization algorithm is a recursive method that takes advantage of the hierarchical nature of the pixelization schemes. The method works as follows:

(i) Start with all the mask polygons that are in pixel i.

(ii) Create polygons for each child pixel of pixel i at the next resolution level.

(iii) Split the mask polygons in pixel i along the child pixel boundaries such that each polygon lies within one child pixel.

(iv) Repeat steps 1-3 for the mask polygons in each child pixel until desired stopping point is reached.

Starting this with pixel $i = 0$, i.e., the whole sky, will pixelize the entire mask in $\mathcal{O}(N \log M)$ time. Thus the pixelization does not add too much overhead time to the overall mask processing.

There are two different methods for choosing the desired stopping point of the pixelization. The simplest method is to stop at a fixed resolution, such that the entire mask is pixelized with pixels of the same size. The example mask from Fig. 2 is shown in the left panel of Fig. 6 pixelized to a fixed resolution in the simple scheme.

Alternatively, the stopping condition can be chosen to be a maximum number of polygons allowed in each pixel: if there are more than $N_{\max}$ polygons in a given pixel, continue the recursion
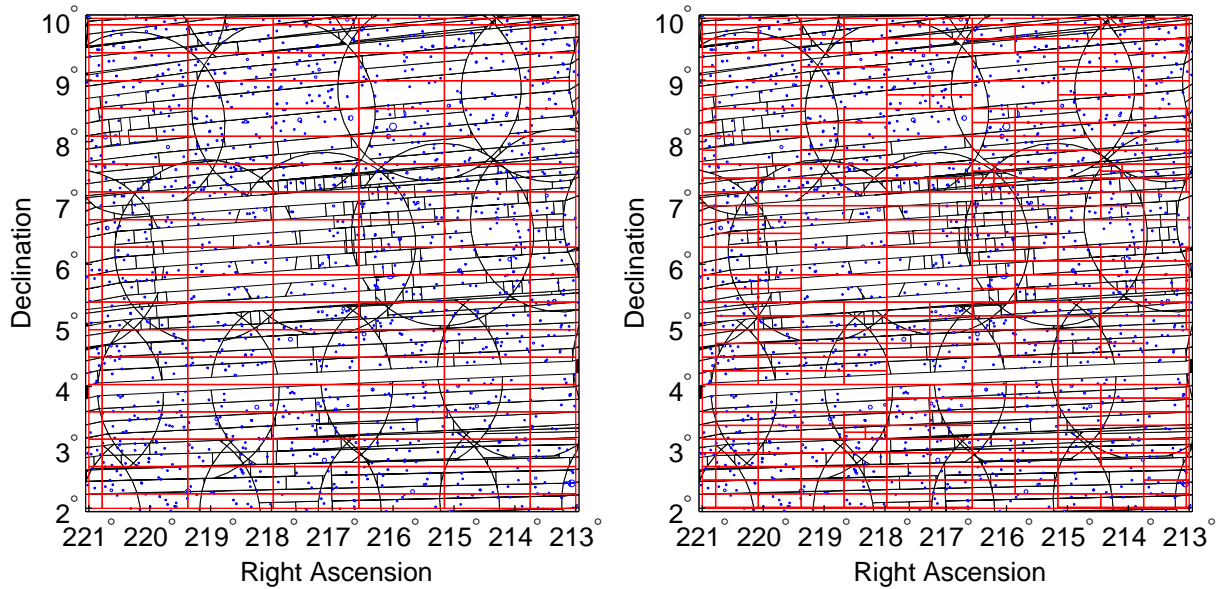
Figure 6. Left: portion of SDSS mask from Fig. 2 pixelized to resolution 8 (4^8 total pixels on the sky) with the simple pixelization scheme. Pixel boundaries are shown in red/light gray. Right: the same mask pixelized with the simple pixelization scheme using the adaptive resolution method with a maximum of 30 polygons per pixel.

and divide those polygons into pixels at the next resolution level. This results in an adaptively pixelized mask, where higher resolutions are automatically used in regions of the mask that are more complicated. This method is especially useful for masks with varying degrees of complexity in different areas. The example mask from Fig. 2 is shown again in the right panel of Fig. 6 pixelized adaptively with $N_{max} = 30$.

The implementation of pixelization in MANGLE allows users to choose either of these methods and to select values for the fixed resolution level or for N_{max}.

3.4 Speed trials

In order to choose optimal default values for the maximum resolution and N_{max} as well as to demonstrate the dramatic improvements in speed that pixelization provides, we have conducted a series of speed trials of the new MANGLE software.

There are two procedures we are interested in optimizing: firstly, basic processing of a mask detailed in §2 involving pixelization, snapping, balkanization, and unification, and secondly, the use of the final mask to identify in which polygon a given point lies. We carried out trials of these two procedures using both the simple and SDSSPix pixelization schemes described in §3.2. For each of these schemes, we tested both the fixed and adaptive resolution methods for stopping the pixelization algorithm described in §3.3. For the fixed resolution method, we measured the time for several different values of the maximum resolution, and for the adaptive resolution method, we measured the time as a function of N_{max}.

The results are shown in Fig. 7. (Note that the overhead time for reading and writing files and doing general setup has been subtracted – the times shown here are just for the primary operations.) From the fixed resolution trials, we see that the optimal resolution choice for the SDSS mask is the one that has approximately 10^5 total pixels on the sky for both the simple and SDSSPix schemes. This corresponds to resolution 9 for the simple scheme and resolu-

tion 6 for SDSSPix. When using the adaptive method, the choice for the maximum number of polygons allowed in each pixel that gives the fastest processing is $N_{max} = 40$ for the simple scheme and $N_{max} = 46$ for SDSSPix. Overall, the fastest choice (by a slight margin) for the SDSS mask is using the SDSSPix scheme with adaptive resolution. It is also interesting to note that different MANGLE processes have different optimum values – for example, snapping is fastest when there are fewer polygons in each pixel compared to balkanization.

The impact of our pixelization algorithm is most clearly demonstrated by Fig. 1. This shows the processing time and polygon identification time for a series of selected portions of the SDSS mask as a function of the total number of polygons, both with and without pixelization – again, overhead time has been subtracted here. Pixelization clearly gives an improvement in speed that becomes increasingly significant for larger numbers of polygons.

To quantify this, we fit theoretical models to each series of trials. Without pixelization, the processing time for snapping, balkanization, and unification is well-fitted by AN^2 where A is a free parameter with a best-fitted value of $A = 3.2 \times 10^{-5}$ CPU seconds. With pixelization (and using the adaptive resolution method with the optimal choice for N_{max}) this is reduced substantially. We fit the times using theoretical models based on the formulas in §3.1 assuming $M \sim N$. We fit the pixelization time with $BN \log_4 N$ and the snapping, balkanization, and unification time with CN where B and C are free parameters. The simple scheme gives $(B, C) = \left(0.0054, \, 4.7 \times 10^{-4}\right)$ and the SDSSPix scheme gives $(B, C) = \left(0.0045, \, 4.8 \times 10^{-4}\right)$ (units are all in CPU seconds) – thus for processing the SDSS mask, the SDSSPix scheme is slightly faster than the simple scheme. Our fits give $C/B = 0.1$, so the overall processing time scales like $\mathcal{O}\left(N + 0.1N \log_4 N\right)$.

These fitted curves are shown in Fig. 1 and allow us to extrapolate estimates for processing masks with larger numbers of polygons. For the full SDSS DR5 mask containing about 300,000 polygons, pixelization reduces the processing time by a factor of
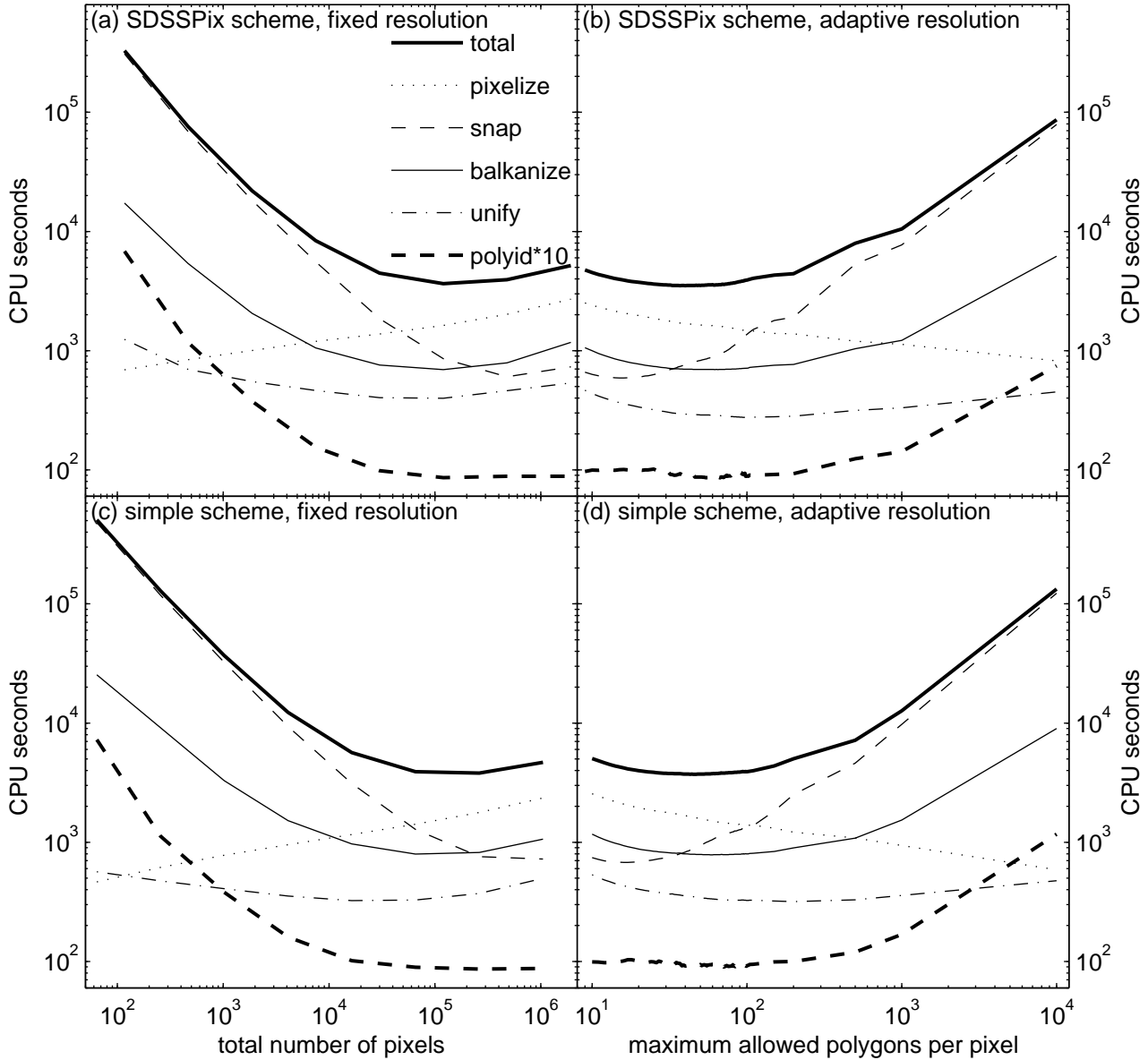
Figure 7. Time required to process the full SDSS DR5 mask with different choices of pixelization schemes and methods. The "total" curves are the sum of the pixelization, snapping, balkanization, and unification curves. Also shown is the time required to identify in which polygon each of the ~400,000 SDSS DR5 galaxies lies (polyid), scaled up by a factor of 10. The plots on the left show time vs. the number of pixels at a fixed resolution. The plots on the right show time vs. N_{max}, the maximum number of polygons in allowed in each pixel when pixelizing with the adaptive resolution method.

~1200. The improvement for future surveys will be even more dramatic: the mask for the co-added LSST survey might contain ~10^8 polygons – pixelization reduces the processing time by a factor of ~24,000.

The time for identifying in which polygon each SDSS galaxy lies is significantly sped up as well: without pixelization it is fitted by DN where $D = 0.012$ CPU seconds, although the scatter is rather large due to dependence on which polygons are used. With pixelization, it is well-fitted by a constant (8.2 CPU seconds for the simple scheme, 7.9 CPU seconds for the SDSSPix scheme) – thus the time for polygon identification does not depend on how many polygons are in the mask if the number of pixels used is chosen to be roughly proportional to the number of polygons. For the SDSS

mask, the time to identify the polygons for all of the SDSS galaxies is reduced by a factor of nearly 500.

4 UNIFICATION WITH *HEALPIX* AND OTHER PIXELIZED TOOLS

HEALPix (Górski et al. 1999b, 2005) is a hierarchical, equal-area, isolatitudinal pixelization scheme for the sphere motivated by computational challenges in analyzing CMB data (Górski et al. 1999a, Bond et al. 1999). Its base resolution consists of 12 equal-area pixels; to generate higher resolutions, each pixel at a given resolution is hierarchically subdivided into four smaller pixels, and it represents an interesting class of spherical projections
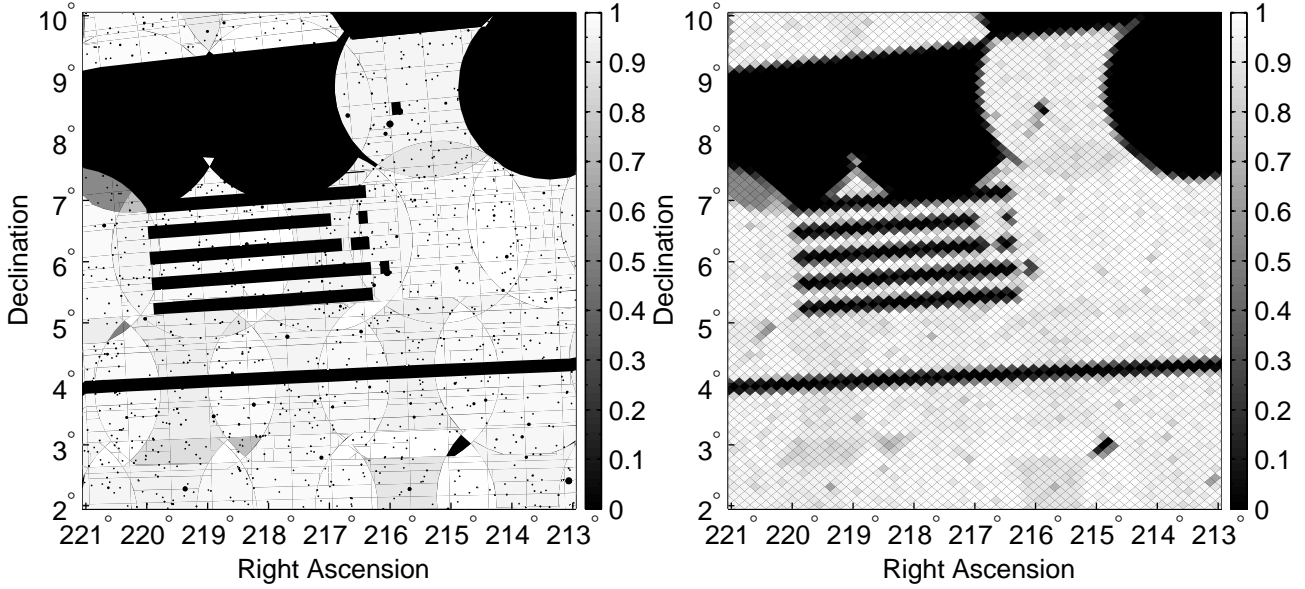
Figure 8. Left: Portion of SDSS mask as shown in Fig. 2. Right: Portion of SDSS mask from Fig. 2 as approximated by *HEALPix* pixels, rasterized to $N_{side} = 512$.

(Calabretta & Roukema 2007). Resolution in *HEALPix* is defined in terms of $N_{\rm side}$, the number of divisions along the side of a base-resolution pixel required to reach the desired resolution. Because of the hierarchical definition of the higher resolution pixels, $N_{\rm side}$ is always a power of 2, and the total number of pixels at a given resolution is $12N_{\rm side}^2$.

HEALPix is a very useful scheme in that it allows for fast and accurate astrophysical computations by means of appropriately discretizing functions on the sphere to high resolution (Wandelt et al. 1998, Doré et al. 2001b). In particular, *HEALPix* includes routines for fast computations of spherical harmonics (Doré, Knox, & Peel 2001a; Hivon et al. 2002; Wandelt et al. 2001; Szapudi et al. 2001). It is widely used in the analysis of cosmic microwave background data from WMAP (Spergel et al. 2007) and has recently been used to approximate galaxy survey masks as well (Percival et al. 2007).

Combining *HEALPix* with MANGLE is useful because it facilitates comparisons between galaxy survey data and the CMB as is done in experiments measuring the integrated Sachs-Wolfe effect (Padmanabhan et al. 2005; Giannantonio et al. 2006; Rassat et al. 2007), the Sunyaev-Zel'dovich effect (Myers et al. 2004; Reid & Spergel 2006), etc., and can also be used for generating masks that block out regions of high dust extinction from galaxy surveys (Schlegel et al. 1998, dust map available in *HEALPix* format online[4]).

In general, it can be applied to any task requiring comparison between a piecewise-constant function on the sphere to a continuous function sampled on a discretized spherical grid. Converting an angular mask into *HEALPix* format also allows for rapid computations of approximate spherical harmonics of the mask using the existing *HEALPix* tools.

The implementation of the *HEALPix* scheme in MANGLE consists of two components:

(i) A new polygon format, "healpix_weight", which allows the user to input a list of weights corresponding to each *HEALPix* pixel at a given $N_{\rm side}$ parameter;

(ii) A new utility, "rasterize", which essentially allows the user to pixelize a mask against the *HEALPix* pixels, by means of a different technique than the pixelization method described in §3.

Together, these new features allow effective two-way conversion between the *HEALPix* specifications and those of MANGLE.

4.1 Importing *HEALPix* maps into MANGLE

The structure of the healpix_weight format is quite simple: an input file consists of a list of numbers corresponding to the weight of each *HEALPix* pixel at a given $N_{\rm side}$ parameter, using the nested numbering scheme described in Górski et al. (2005). In addition, the definition of the $N_{\rm side}$ parameter is extended to include 0, which corresponds to a single pixel covering the entire sphere. MANGLE constructs polygons approximately equivalent to the *HEALPix* pixels through the following procedure:

(i) The exact azimuth and elevation of each vertex of a given pixel are calculated using the *HEALPix* utility "pix2vec_nest";

(ii) The exact azimuth and elevation of each vertex of the four child pixels of the current pixel are calculated using the same utility, four of which are the midpoints of the edges of the current pixel;

(iii) The four vertices of the given pixel are combined with the four midpoints to construct the current pixel. Each edge is defined by the circle that passes through the two vertices and the midpoint, using the "edges" format described in Hamilton & Tegmark (2004).

(iv) To eliminate spurious antipodal pieces of the polygon defining the pixel, a fifth cap is constructed whose axis coordinates are the exact centre of the current pixel and whose radius is 10^{-6} radians greater than the distance from the centre of the pixel to any of the four vertices—this cap thus encloses the entire pixel.

[4] http://lambda.gsfc.nasa.gov/product/foreground/ebv_maps.cfm
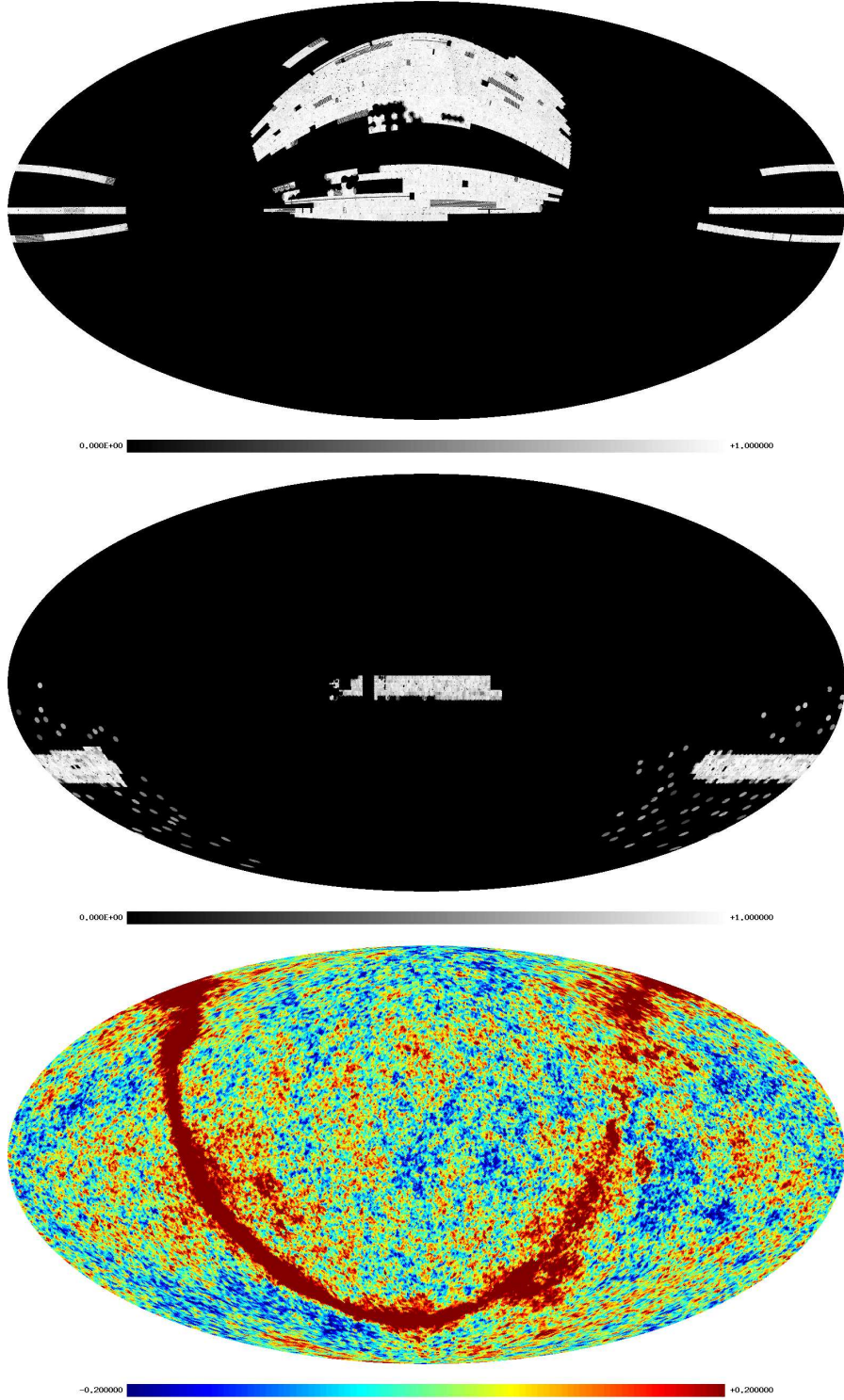
Figure 9. The new version of MANGLE can output angular masks in *HEALPix* format, which allows for easy comparisons to CMB and other sky map data and allows users to take advantage of existing *HEALPix* tools. Top: The SDSS DR5 completeness mask, rasterized and plotted using HEALPix routines. Middle: The final 2dFGRS completeness mask as determined in Hamilton et al. (2008), rasterized and plotted using *HEALPix* routines. Bottom: CMB temperature difference map measured by WMAP channel 4, with units in mK. Spergel et al. (2007) All three maps are shown at $N_{side} = 512$ in celestial coordinates.

A similar technique could be applied to incorporate other pixelization schemes not exactly described by spherical polygons as well. It is important to note that the pixels constructed in this manner are not exactly equivalent to the actual *HEALPix* pixels – while the hierarchical and isolatitudinal properties of the *HEALPix* pixels are preserved, the equal area property is not. For example, at $N_{\mathrm{side}} = 1$, the areas of the approximate pixels differ on average from the actual area by about 0.08%.

However, as the N_{side} parameter increases, this difference decreases rapidly: at $N_{\mathrm{side}} = 512$, the average difference from the actual area is 0.000002%. The boundaries of the both the actual *HEALPix* pixels and our circles approximating them become straight lines in the flat-sky approximation, i.e., when the pixel size is much less than 1 radian. Thus for the resolutions at which *HEALPix* is typically used, the difference is totally negligible.

In applications involving integrations over the sphere, such as calculating spherical harmonics, the slight area differences between the pixels can be corrected for in a straightforward manner since the area of each pixel is known (and can be easily extracted using MANGLE's "area" format): simply multiply the value of the function in each pixel by the area of that pixel divided by the average pixel area, and then the *HEALPix* spherical harmonics routines will be exact. Furthermore, the paranoid user can get higher precision by rasterizing to a higher resolution and then using the "ud_grade" *HEALPix* utility to obtain results for lower resolutions.

4.2 Exporting polygon files as *HEALPix* maps

The idea behind the rasterization method used to convert polygon files into *HEALPix* maps is very similar to that of pixelization: to split up the polygons that comprise a mask using a given set of pixels, such that afterward each polygon lies in only one pixel. However, rasterization is somewhat different, in that afterward the converse statement also holds: each pixel contains only one polygon, namely, itself. In particular, rasterization uses an arbitrary user-defined spherical pixelization as the pre-determined scheme against which to split up the polygons in a given input mask. In general, the user-defined "rasterizer" pixels may be from any pixelization of the sphere, but the method was originally developed for use with the approximate *HEALPix* pixels described in the previous section. From a practical standpoint, it would be simplest to implement any spherical pixelization that can be exactly represented as spherical polygons by following the steps in §3.2.3; however, for pixelization schemes that do not possess this property (such as *HEALPix*), rasterization provides an alternate method of implementation.

The final product of rasterization is a polygon file in which each polygon corresponds to one of the rasterizer pixels (either the approximate *HEALPix* pixels or a user-defined set of pixels) and its weight is the area-averaged weight of the input angular mask within that pixel. This output can easily be converted into a FITS file read by the *HEALPix* software using a simple script provided with MANGLE.

Rasterization consists of the following steps:

(i) Calculate the area of each rasterizer pixel;

(ii) Compute the area of the intersection of each input mask polygon with each rasterizer (e.g., *HEALPix*) pixel;

(iii) Calculate the area-averaged weight within each rasterizer pixel.

Note that, as with snapping, balkanization, and unification, this procedure is greatly accelerated by pixelizing both the rasterizer polygons and the polygons defining the mask to the same resolution

using one of the pixelization schemes described in §3.2, e.g. simple or SDSSPix. Then step (ii) in the above procedure then involves comparing only polygons in the same simple/SDSSPix pixel. Our example mask from Fig. 2 (duplicated in the left panel of Fig. 8) is shown in the right panel of Fig. 8 rasterized with *HEALPix* pixels at $N_{\mathrm{side}} = 512$. The ability to convert between MANGLE and *HEALPix* formats allows for straightforward comparisons between different types of functions defined on the sphere, e.g. angular masks of galaxy surveys and the CMB, as illustrated in Fig. 9.

5 SUMMARY

As technologies for surveying the sky continue to improve, the process of managing the angular masks of galaxy surveys grows ever more complicated. The primary purpose of this paper has been to present a set of dramatically faster algorithms for completing these tasks.

These algorithms are based on dividing the sky into regions called "pixels" and performing key operations only within each pixel rather than across the entire sky. The pixelization is based on a hierarchical subdivision of the sky – this produces a quadtree data structure that keeps track of which polygons are nearby each other. The preprocessing step of pixelization is $\mathcal{O}\left(N \log N\right)$ for a mask with N polygons and $\sim N$ pixels, and it reduces the mask processing time from $\mathcal{O}\left(N^2\right)$ to $\mathcal{O}\left(N\right)$. Furthermore, it reduces the time required to locate a point within a polygon from $\mathcal{O}\left(N\right)$ to $\mathcal{O}\left(1\right)$.

This method is exact, i.e., it does not make a discrete approximation to the mask, and it takes only a tiny fraction of the computation time. It accelerates the processing of the SDSS mask by a factor of about 1200 and reduces the time to locate all of the galaxies in the SDSS mask by a factor of nearly 500. It will provide even more dramatic gains for future surveys: processing time for the LSST large-scale structure mask could be reduced by a factor of about 24,000.

We have also described a method for converting between masks described by spherical polygons and sky maps in the *HEALPix* format commonly used by CMB and large-scale structure experiments. This provides a convenient way to work with both piecewise-constant functions on the sky such as the completeness of a galaxy survey and continuously varying sky maps such as the CMB temperature. Converting angular masks into *HEALPix* format also allows users to take advantage of existing *HEALPix* tools for rapidly computing spherical harmonics.

All of the new algorithms and features detailed here have been integrated into the MANGLE software suite, which is available for free download at `http://space.mit.edu/home/tegmark/mangle/`. This updated software package should prove increasingly useful in the coming years, especially as next-generation surveys such as DES, WFMOS, Pan-STARRS, and LSST get underway.

References

Adelman-McCarthy J. K., et al., 2007, ApJS, 172, 634

Banerji M., Abdalla F. B., Lahav O., Lin H., 2008, MNRAS, 386, 1219

Blanton M. R., Schlegel D. J., Strauss M. A., Brinkmann J., Finkbeiner D., Fukugita M., Gunn J. E., Hogg D. W., Ivezić Ž., Knapp G. R., Lupton R. H., Munn J. A., Schneider D. P., Tegmark M., Zehavi I., 2005, AJ, 129, 2562

Bond J. R., Crittenden R. G., Jaffe A. H., Knox L., 1999, Comput. Sci. Eng., Vol. 1, No. 2, p. 21 - 35, 1, 21

Calabretta M. R., Roukema B. F., 2007, MNRAS, 381, 865

Colless M., et al., 2001, MNRAS, 328, 1039

Colless M., et al., 2003, astro-ph/0306581

Collister A., Lahav O., Blake C., Cannon R., Croom S., Drinkwater M., Edge A., Eisenstein D., Loveday J., Nichol R., Pimbblet K., de Propris R., Roseboom I., Ross N., Schneider D. P., Shanks T., Wake D., 2007, MNRAS, 375, 68

Conroy C., Coil A. L., White M., Newman J. A., Yan R., Cooper M. C., Gerke B. F., Davis M., Koo D. C., 2005, ApJ, 635, 990

da Costa L. N., Pellegrini P. S., Davis M., Meiksin A., Sargent W. L. W., Tonry J. L., 1991, ApJS, 75, 935

Dark Energy Survey Collaboration 2005, astro-ph/0510346

de Berg M., van Kreveld M., Overmars M., Schwarzkopf O., 2000, Computational Geometry: Algorithms and Applications. Springer, New York, pp 291–298

Doré O., Knox L., Peel A., 2001a, Phys. Rev. D, 64, 083001

Doré O., Teyssier R., Bouchet F. R., Vibert D., Prunet S., 2001b, A&A, 374, 358

Giannantonio T., Crittenden R. G., Nichol R. C., Scranton R., Richards G. T., Myers A. D., Brunner R. J., Gray A. G., Connolly A. J., Schneider D. P., 2006, Phys. Rev. D, 74, 063520

Glazebrook K., Eisenstein D., Dey A., Nichol R., 2006, WFMOS DETF White Paper, http://home.fnal.gov/~rocky/DETF/Dey.pdf

Goodman J. E., O'Rourke J., eds, 2004, Handbook of Discrete and Computational Geometry. Chapman & Hall/CRC, New York, pp 778, 871–872

Górski K. M., et al., 1999a Evolution of Large Scale Structure : From Recombination to Garching. Proceedings of the MPA-ESO Cosmology Conference, Garching, Germany, pp 37–+

Górski K. M., Hivon E., Banday A. J., Wandelt B. D., Hansen F. K., Reinecke M., Bartelmann M., 2005, ApJ, 622, 759

Górski K. M., Wandelt B. D., Hansen F. K., Hivon E., Banday A. J., 1999b, astro-ph/9905275

Hamilton A. J. S., 1993a, ApJ, 406, L47

Hamilton A. J. S., 1993b, ApJ, 417, 19

Hamilton A. J. S., Tegmark M., 2004, MNRAS, 349, 115

Hamilton A. J. S., Tegmark M., Swanson M. E. C., 2008, in preparation

Hikage C., Matsubara T., Suto Y., Park C., Szalay A. S., Brinkmann J., 2005, PASJ, 57, 709

Hivon E., Górski K. M., Netterfield C. B., Crill B. P., Prunet S., Hansen F., 2002, ApJ, 567, 2

Huchra J., Davis M., Latham D., Tonry J., 1983, ApJS, 52, 89

Kaiser N., 2004, in Oschmann Jr. J. M., ed., Proc. SPIE Vol. 5489, Ground-based Telescopes. International Society for Optical Engineering, Bellingham, WA, pp 11–22

LSST Collaboration 2006, LSST DETF White Paper, http://www.lsst.org/Science/docs/LSST_DETF_Whitepaper.pdf

Maddox S. J., Efstathiou G., Sutherland W. J., 1990a, MNRAS, 246, 433

Maddox S. J., Efstathiou G., Sutherland W. J., 1996, MNRAS, 283, 1227

Maddox S. J., Efstathiou G., Sutherland W. J., Loveday J., 1990b, MNRAS, 243, 692

Mandelbaum R., Hirata C. M., Seljak U., Guzik J., Padmanabhan N., Blake C., Blanton M. R., Lupton R., Brinkmann J., 2005, MNRAS, 361, 1287

Myers A. D., Shanks T., Outram P. J., Frith W. J., Wolfendale A. W., 2004, MNRAS, 347, L67

Nishimichi T., Kayo I., Hikage C., Yahata K., Taruya A., Jing Y. P., Sheth R. K., Suto Y., 2007, PASJ, 59, 93

Oyaizu H., Lima M., Cunha C. E., Lin H., Frieman J., Sheldon E. S., 2008, ApJ, 674, 768

Padmanabhan N., Hirata C. M., Seljak U., Schlegel D. J., Brinkmann J., Schneider D. P., 2005, Phys. Rev. D, 72, 043525

Park C., Choi Y.-Y., Vogeley M. S., Gott J. R. I., Blanton M. R., 2007, ApJ, 658, 898

Park C., Choi Y.-Y., Vogeley M. S., Gott J. R. I., Kim J., Hikage C., Matsubara T., Park M.-G., Suto Y., Weinberg D. H., 2005, ApJ, 633, 11

Percival W. J., Nichol R. C., Eisenstein D. J., Frieman J. A., Fukugita M., Loveday J., Pope A. C., Schneider D. P., Szalay A. S., Tegmark M., Vogeley M. S., Weinberg D. H., Zehavi I., Bahcall N. A., Brinkmann J., Connolly A. J., Meiksin A., 2007, ApJ, 657, 645

Rassat A., Land K., Lahav O., Abdalla F. B., 2007, MNRAS, 377, 1085

Reid B. A., Spergel D. N., 2006, ApJ, 651, 643

Saunders W., Sutherland W. J., Maddox S. J., Keeble O., Oliver S. J., Rowan-Robinson M., McMahon R. G., Efstathiou G. P., Tadros H., White S. D. M., Frenk C. S., Carramiñana A., Hawkins M. R. S., 2000, MNRAS, 317, 55

Schlegel D. J., Finkbeiner D. P., Davis M., 1998, ApJ, 500, 525

Shen Y., Strauss M. A., Oguri M., Hennawi J. F., Fan X., Richards G. T., Hall P. B., Gunn J. E., Schneider D. P., Szalay A. S., Thakar A. R., Vanden Berk D. E., Anderson S. F., Bahcall N. A., Connolly A. J., Knapp G. R., 2007, AJ, 133, 2222

Spergel D. N., et al., 2007, ApJS, 170, 377

Stoughton C., et al., 2002, AJ, 123, 485

Strauss M. A., Huchra J. P., Davis M., Yahil A., Fisher K. B., Tonry J., 1992, ApJS, 83, 29

Stubbs C. W., Sweeney D., Tyson J. A., the LSST Collaboration 2004, in Bulletin of the American Astronomical Society Vol. 36. pp 1527–+

Swanson M. E. C., Tegmark M., Blanton M., Zehavi I., 2008, MNRAS, 385, 1635

Szapudi I., Prunet S., Colombi S., 2001, ApJ, 561, L11

Tegmark M., et al., 2004, ApJ, 606, 702

Tegmark M., et al., 2006, Phys. Rev. D, 74, 123507

Tegmark M., Hamilton A. J. S., Xu Y., 2002, MNRAS, 335, 887

Tinker J. L., Conroy C., Norberg P., Patiri S. G., Weinberg D. H., Warren M. S., 2007, arXiv:0707.3445

Tyson J. A., 2002, in Tyson J. A., Wolff S., eds, Proc. SPIE Vol. 4836, Survey and Other Telescope Technologies and Discoveries. International Society for Optical Engineering, Bellingham, WA, pp 10–20

Tyson J. A., 2006, in Liss T. M., ed., AIP Conf. Ser. Vol. 870, Intersections of Particle and Nuclear Physics: 9th Conference CIPAN2006. Am. Inst. Phys, New York, pp 44–52

Wandelt B. D., Hivon E., Gorski K. M., 1998, astro-ph/9803317

Wandelt B. D., Hivon E., Górski K. M., 2001, Phys. Rev. D, 64, 083003

Wang Y., Yang X., Mo H. J., van den Bosch F. C., 2007, ApJ, 664, 608

Yamamoto K., Bassett B. A., Nichol R. C., Suto Y., Yahata K., 2006, Phys. Rev. D, 74, 063525

York D. G., et al., 2000, AJ, 120, 1579